UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/917,312	07/27/2001	Hown Cheng	Stream-09US	9556

7590 05/20/2004

Maryam Imam, Esq.
LAW OFFICES OF IMAM
Suite 1010
111 North Market Street
San Jose, CA 95113

EXAMINER	
GERSTL, SHANE F	

ART UNIT	PAPER NUMBER
2183	9

DATE MAILED: 05/20/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/917,312	CHENG ET AL.
	Examiner	Art Unit
	Shane F Gerstl	2183

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *29 January 2003 and 27 July 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-26* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-26* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *27 July 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-26 have been examined.

Papers Received

2. Receipt is acknowledged of 37 CFR 1.137(b) petition, drawings, and extension of time papers submitted, where the papers have been placed of record in the file.

Specification

3. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

The following title is suggested: ARCHITECTURE FOR HARDWARE-ASSISTED CONTEXT SWITCHING BETWEEN REGISTER GROUPS DEDICATED TO TIME-CRITICAL OR NON-TIME-CRITICAL TASKS WITHOUT SAVING STATE.

Duplicate Claims

4. Applicant is advised that should claims 1 and 13 be found allowable, claims 14 and 26 will be objected to under 37 CFR 1.75 as being a substantial duplicate thereof. When two claims in an application are duplicates or else are so close in content that they both cover the same thing, despite a slight difference in wording, it is proper after allowing one claim to object to the other as being a substantial duplicate of the allowed claim. See MPEP § 706.03(k). In this case, the only difference in the effectively identical claim sets (1/14 and 13/26) is that the first set of registers changes from being used for executing time-critical tasks to non-timing-critical tasks and the second set of registers changes from being used for executing non-timing-critical tasks to timing-critical tasks. This is effectively only a changing of names (which is not patentable)

since the terms "first" and "second" are viewed simply as names and do not infer any type of structure in apparatus or method.

Claim Objections

5. Claims 8 and 21 are objected to because of the following informalities: A claim which depends from a dependent claim should not be separated by any claim which does not also depend from said dependent claim. It should be kept in mind that a dependent claim may refer to any preceding independent claim. In general, applicant's sequence will not be changed. See MPEP § 608.01(n). Appropriate correction is required.

Claim Rejections - 35 USC § 102

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

7. Claims 1-10, 13-23, and 26 are rejected under 35 U.S.C. 102(b) as being anticipated by Parady (5,933,627).

8. In regard to claim 1, Parady discloses a processing system (figure 3) for processing information efficiently and cost-effectively by switching between execution of time-critical and non-time-critical tasks comprising:

a. a processing unit (figure 1, element 41)

b. a first register group coupled to said processing unit and including a first set of registers (figure 3, element 48, thread 0), said first register group for

updating the status of said first set of registers, said processing unit reading the status of said first set of registers to execute time-critical tasks; Figure 3 shows that the processing unit (execution units) read the status of the registers. Column 2, lines 18-20 show that thread switching is done (to and from thread 0) based on long-latency events or time-critical tasks. Thus, each thread executes time-critical tasks that cause switching at some point.

c. and a second register group coupled to said processing unit and including a second set of registers (figure 3, element 48, thread 1), said second register group for updating the status of said second set of registers, said processing unit reading the status of said second set of registers to execute non-time-critical tasks, said processing unit switching to execute non-time-critical tasks by avoiding saving the status of said first set of registers; Figure 3 shows that the processing unit (execution units) read the status of the registers. Column 2, lines 18-20 show that thread switching is done (to and from thread 0) based on long-latency events or time-critical tasks. All other instructions are non-time-critical events. Thus, each thread executes non-time-critical tasks before switching occurs.

d. wherein said processing unit switches between executing time-critical tasks and non-time-critical tasks efficiently and cost-effectively by avoiding saving status of the first or second set of registers. As shown in the background, the switching is done to switch from a long-latency event (load, store, jump, etc as shown in column 4, lines 5-8) so that processing can continue and the

processor isn't held up. Thus, switching generally is done to execute a non-time-critical set of tasks. In some instances, as shown in column 4, lines 46-50, a return is made to the long-latency or time-critical task when the data needed is received; this is a switch to a time-critical task. Column 3, lines 54-56 shows that on a thread switch, instructions simply pick up where they left off. There is no saving of status mentioned in the disclosure of Parady; the different registers facilitate the switching with their data for each thread so instructions can immediately begin on a switch.

9. In regard to claim 2, Parady discloses a processing system as recited in claim 1 wherein said processing unit switches to executing said time-critical tasks by avoiding saving the status of said second set of registers, as shown above.

10. In regard to claim 3, Parady discloses a processing system as recited in claim 1 further including a code random access memory (RAM) for storing instructions for execution of said time-critical tasks, said processing unit fetching instructions from said code RAM to execute said time-critical tasks. Column 2, lines 66-67 show that figure 1 is the base model for incorporation of the invention of figure 3. Figure 1 shows an instruction cache (element 12) that stores all instructions, including the time-critical instructions or tasks, which are then fetched, decoded and dispatched for execution. It is inherent that a cache is a RAM since any position may be accessed at one time (random access) based on a tag address.

11. In regard to claim 4, Parady discloses a processing system as recited in claim 1 further including an instruction cache (I-cache) for storing instructions for execution of

said non-time-critical tasks, said processing unit fetching instructions from said instruction cache to execute said non-time-critical tasks. Column 2, lines 66-67 show that figure 1 is the base model for incorporation of the invention of figure 3. Figure 1 shows an instruction cache (element 12) that stores all instructions, including the non-time-critical instructions or tasks, which are then fetched, decoded and dispatched for execution.

12. In regard to claim 5, Parady discloses a processing system as recited in claim 1 further including a first data memory for storing data for executing said time-critical tasks, said processing system further including a second data memory for storing data for executing said non-time-critical tasks. The registers of mention are in themselves data memories and hold data for execution of both types of tasks.

13. In regard to claim 6, Parady discloses a processing system as recited in claim 1 further including a high priority interrupt controller responsive to interrupt commands requesting service, said high priority interrupt controller signaling said processing unit to provide service by executing said time-critical tasks. As shown above, there is a switch on a time-critical task to a non-time-critical task with one interrupt (low priority) and a switch back to the time-critical task with another interrupt (high priority). There are inherently controllers for these interrupts.

14. In regard to claim 7, Parady discloses a processing system as recited in claim 1 further including a low priority interrupt controller responsive to interrupt commands requesting service, said low priority interrupt controller signaling said processing unit to provide service by executing said non-time-critical tasks. As shown above, there is a

switch on a time-critical task to a non-time-critical task with one interrupt (low priority) and a switch back to the time-critical task with another interrupt (high priority). There are inherently controllers for these interrupts.

15. In regard to claim 8, Parady discloses a processing system as recited in claim 5 further including a data bus bridge for transferring data from an external memory to said second data memory. Figure 1 (which is shown to be the base processor for the invention in column 2, lines 66-67) shows that the data memories (registers) have a data bridge (cache control, element 22) for transferring data from an external memory (figure 2).

16. In regard to claim 9, Parady discloses a processing system as recited in claim 1 wherein said processing system is in communication with a communication system, said processing system further including a register bus (R-bus) bridge for providing an interface between said processing system and said communication system. Figure 1 (which is shown to be the base processor for the invention in column 2, lines 66-67) shows that the data memories (registers) have a register bus bridge (cache control/system interface, element 22) for providing an interface and transferring data from an external memory (figure 2) to the registers. Since the processing system and the external cache system communicate with each other they are both communication systems and thus the processing system communicates with a communication system.

17. In regard to claim 10, Parady discloses a processing system as recited in claim 1 further including a real-time operating system (RTOS) for providing services for execution of said non-time-critical tasks. Column 4, lines 32-41 show an embodiment

where the processing system is used for an operating system. The enclosed IEEE standard definition of "operating system" states that an operating system controls execution of a computer program. This means that the operating system given by Parady controls the execution of (or provides services for) all program instructions including non-time-critical tasks. The included IEEE standard definition of "real time system" shows that this is a system where both the computational result and the time at which they are computed and output are important. The pipelined superscalar processor of Parady is inherently dependent on computational output time so that instructions are properly scheduled for execution and thus the operating system is inherently a real-time operating system that controls this execution.

18. In regard to claim 13, Paraday discloses a method for processing information efficiently and cost-effectively by switching between execution of time-critical tasks and non-time-critical tasks comprising:

a. updating the status of a first set of registers within a first register group; Figure 3, element 48 shows a first group of registers for thread 0 that are updated by the results from the execution units.

b. reading the status of the first set of registers; Figure 3 shows that the execution units read the status of the registers.

c. executing time-critical tasks, Column 2, lines 18-20 show that thread switching is done (to and from thread 0) based on long-latency events or time-critical tasks. Thus, each thread executes time-critical tasks that cause switching at some point.

d. updating the status of a second set of registers within a second register group; Figure 3, element 48 shows a second group of registers for thread 1 that are updated by the results from the execution units.

e. reading the status of the second set of registers, Figure 3 shows that the execution units read the status of the registers.

f. executing non-time-critical tasks; Column 2, lines 18-20 show that thread switching is done (to and from thread 0) based on long-latency events or time-critical tasks. All other instructions are non-time-critical events. Thus, each thread executes non-time-critical tasks before switching occurs.

g. and switching to execute non-time-critical tasks by avoiding saving the status of the first set of registers. As shown in the background, the switching is done to switch from a long-latency event (load, store, jump, etc as shown in column 4, lines 5-8) so that processing can continue and the processor isn't held up. Thus, switching generally is done to execute a non-time-critical set of tasks. In some instances, as shown in column 4, lines 46-50, a return is made to the long-latency or time-critical task when the data needed is received; this is a switch to a time-critical task. Column 3, lines 54-56 shows that on a thread switch, instructions simply pick up where they left off. There is no saving of status mentioned in the disclosure of Parady; the different registers facilitate the switching with their data for each thread so instructions can immediately begin on a switch.

19. In regard to claim 14, Parady discloses a processing system (figure 3) for processing information efficiently and cost-effectively by switching between execution of time-critical and non-time-critical tasks comprising:

a. a processing unit (figure 1, element 41)

b. a first register group coupled to said processing unit and including a first set of registers (figure 3, element 48, thread 0), said first register group for updating the status of said first set of registers, said processing unit reading the status of said first set of registers to execute non-time-critical tasks; Figure 3 shows that the processing unit (execution units) read the status of the registers. Column 2, lines 18-20 show that thread switching is done (to and from thread 0) based on long-latency events or time-critical tasks. All other instructions are non-time-critical events. Thus, each thread executes non-time-critical tasks before switching occurs.

c. and a second register group coupled to said processing unit and including a second set of registers (figure 3, element 48, thread 1), said second register group for updating the status of said second set of registers, said processing unit reading the status of said second set of registers to execute time-critical tasks, said processing unit switching to execute time-critical tasks by avoiding saving the status of said first set of registers; Figure 3 shows that the processing unit (execution units) read the status of the registers. Column 2, lines 18-20 show that thread switching is done (to and from thread 0) based on long-latency events

or time-critical tasks. Thus, each thread executes time-critical tasks that cause switching at some point.

d. wherein said processing unit switches between executing time-critical tasks and non-time-critical tasks efficiently and cost-effectively by avoiding saving status of the first or second set of registers. As shown in the background, the switching is done to switch from a long-latency event (load, store, jump, etc as shown in column 4, lines 5-8) so that processing can continue and the processor isn't held up. Thus, switching generally is done to execute a non-time-critical set of tasks. In some instances, as shown in column 4, lines 46-50, a return is made to the long-latency or time-critical task when the data needed is received; this is a switch to a time-critical task. Column 3, lines 54-56 shows that on a thread switch, instructions simply pick up where they left off. There is no saving of status mentioned in the disclosure of Parady; the different registers facilitate the switching with their data for each thread so instructions can immediately begin on a switch.

20. In regard to claim 15, Parady discloses a processing system as recited in claim 14 wherein said processing unit switches to executing said non-time-critical tasks by avoiding saving the status of said second set of registers, as shown above.

21. In regard to claim 16, Parady discloses a processing system as recited in claim 14 further including a code random access memory (RAM) for storing instructions for execution of said time-critical tasks, said processing unit fetching instructions from said code RAM to execute said time-critical tasks. Column 2, lines 66-67 show that figure 1

is the base model for incorporation of the invention of figure 3. Figure 1 shows an instruction cache (element 12) that stores all instructions, including the time-critical instructions or tasks, which are then fetched, decoded and dispatched for execution. It is inherent that a cache is a RAM since any position may be accessed at one time (random access) based on a tag address.

22. In regard to claim 17, Parady discloses a processing system as recited in claim 14 further including an instruction cache (I-cache) for storing instructions for execution of said non-time-critical tasks, said processing unit fetching instructions from said instruction cache to execute said non-time-critical tasks. Column 2, lines 66-67 show that figure 1 is the base model for incorporation of the invention of figure 3. Figure 1 shows an instruction cache (element 12) that stores all instructions, including the non-time-critical instructions or tasks, which are then fetched, decoded and dispatched for execution.

23. In regard to claim 18, Parady discloses a processing system as recited in claim 14 further including a first data memory for storing data for executing said time-critical tasks, said processing system further including a second data memory for storing data for executing said non-time-critical tasks. The registers of mention are in themselves data memories and hold data for execution of both types of tasks.

24. In regard to claim 19, Parady discloses a processing system as recited in claim 14 further including a high priority interrupt controller responsive to interrupt commands requesting service, said high priority interrupt controller signaling said processing unit to provide service by executing said time-critical tasks. As shown above, there is a switch

on a time-critical task to a non-time-critical task with one interrupt (low priority) and a switch back to the time-critical task with another interrupt (high priority). There are inherently controllers for these interrupts.

25. In regard to claim 20, Parady discloses a processing system as recited in claim 14 further including a low priority interrupt controller responsive to interrupt commands requesting service, said low priority interrupt controller signaling said processing unit to provide service by executing said non-time-critical tasks. As shown above, there is a switch on a time-critical task to a non-time-critical task with one interrupt (low priority) and a switch back to the time-critical task with another interrupt (high priority). There are inherently controllers for these interrupts.

26. In regard to claim 21, Parady discloses a processing system as recited in claim 18 further including a data bus bridge for transferring data from an external memory to said second data memory. Figure 1 (which is shown to be the base processor for the invention in column 2, lines 66-67) shows that the data memories (registers) have a data bridge (cache control, element 22) for transferring data from an external memory (figure 2).

27. In regard to claim 22, Parady discloses a processing system as recited in claim 14 wherein said processing system is in communication with a communication system, said processing system further including a register bus (R-bus) bridge for providing an interface between said processing system and said communication system. Figure 1 (which is shown to be the base processor for the invention in column 2, lines 66-67) shows that the data memories (registers) have a register bus bridge (cache

control/system interface, element 22) for providing an interface and transferring data from an external memory (figure 2) to the registers. Since the processing system and the external cache system communicate with each other they are both communication systems and thus the processing system communicates with a communication system.

28. In regard to claim 23, Parady discloses a processing system as recited in claim 14 further including a real-time operating system (RTOS) for providing services for execution of said non-time-critical tasks. Column 4, lines 32-41 show an embodiment where the processing system is used for an operating system. The enclosed IEEE standard definition of "operating system" states that an operating system controls execution of a computer program. This means that the operating system given by Parady controls the execution of (or provides services for) all program instructions including non-time-critical tasks. The included IEEE standard definition of "real time system" shows that this is a system where both the computational result and the time at which they are computed and output are important. The pipelined superscalar processor of Parady is inherently dependent on computational output time so that instructions are properly scheduled for execution and thus the operating system is inherently a real-time operating system that controls this execution.

29. In regard to claim 26, Paraday discloses a method for processing information efficiently and cost-effectively by switching between execution of time-critical tasks and non-time-critical tasks comprising:

a. updating the status of a first set of registers within a first register group; Figure 3, element 48 shows a first group of registers for thread 0 that are updated by the results from the execution units.

b. reading the status of the first set of registers; Figure 3 shows that the execution units read the status of the registers.

c. executing non-time-critical tasks, Column 2, lines 18-20 show that thread switching is done (to and from thread 0) based on long-latency events or time-critical tasks. All other instructions are non-time-critical events. Thus, each thread executes non-time-critical tasks before switching occurs.

d. updating the status of a second set of registers within a second register group; Figure 3, element 48 shows a second group of registers for thread 1 that are updated by the results from the execution units.

e. reading the status of the second set of registers, Figure 3 shows that the execution units read the status of the registers.

f. executing non-time-critical tasks; Column 2, lines 18-20 show that thread switching is done (to and from thread 0) based on long-latency events or time-critical tasks. Thus, each thread executes time-critical tasks that cause switching at some point.

g. and switching to execute time-critical tasks by avoiding saving the status of the first set of registers. As shown in the background, the switching is done to switch from a long-latency event (load, store, jump, etc as shown in column 4, lines 5-8) so that processing can continue and the processor isn't held up. Thus,

switching generally is done to execute a non-time-critical set of tasks. In some instances, as shown in column 4, lines 46-50, a return is made to the long-latency or time-critical task when the data needed is received; this is a switch to a time-critical task. Column 3, lines 54-56 shows that on a thread switch, instructions simply pick up where they left off. There is no saving of status mentioned in the disclosure of Parady; the different registers facilitate the switching with their data for each thread so instructions can immediately begin on a switch.

Claim Rejections - 35 USC § 103

30. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

31. Claims 11-12 and 24-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Parady in view of Kiriyama (5,561,466).

32. In regard to claim 11,

a. Parady discloses a processing system as recited in claim 1. Parady has shown in column 2, lines 66-67 that the invention is implemented on an UltraSparc microprocessor. The included Tremblay reference is a document on the structure of the UltraSparc processor and shows in section VII, which starts on page 1661, that there is multimedia support and thus support for video and

audio and thus the disclosure of Parady inherently has support for audio and video processing.

b. Parady does not explicitly disclose wherein said processing system is employed in an audio and video encoder/decoder (codec), said audio and video codec performing compression of audio data and video data to generate a compressed audio stream and a compressed video stream.

c. Kiriyama has disclosed an audio/video multiplexer and demultiplexer system (title) that encodes video and audio (figure 5, elements 37 and 45) as well as decodes video and audio (figure 8, elements 73 and 79). Column 4, lines 37-43 show that the video encoding uses data compression in an embodiment. Column 4, lines 60-64 show that the audio data is digitized in the same manner as the video and thus in an embodiment can also be compressed.

d. Column 1, line 63 – column 2, line 5 show that the invention of Kiriyama achieves excellent efficiency and allows reproduction of synchronous audio and video data so that there is no audio-visual gap in this data. This efficiency and audio/video synchronization would have motivated one of ordinary skill in the art to modify the design of Parady to manipulate audio and video data and tasks as taught by Kiriyama.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Parady to manipulate audio and video data and tasks as taught by Kiriyama so that excellent efficiency and synchronous audio and video data may be realized.

33. In regard to claim 12,

a. Parady discloses a processing system as recited in claim 1. Parady has shown in column 2, lines 66-67 that the invention is implemented on an UltraSparc microprocessor. The included Tremblay reference is a document on the structure of the UltraSparc processor and shows in section VII, which starts on page 1661, that there is multimedia support and thus support for video and audio and thus the disclosure of Parady inherently has support for audio and video processing.

b. Parady does not explicitly disclose wherein said non-time-critical tasks include multiplexing of said compressed audio stream with said compressed video stream, said time-critical tasks including providing video data for compression thereof.

c. Kiriyama has disclosed an audio/video multiplexer and demultiplexer system (title) that encodes video and audio (figure 5, elements 37 and 45) as well as decodes video and audio (figure 8, elements 73 and 79). Column 4, lines 37-43 show that the video encoding uses data compression in an embodiment. Column 4, lines 60-64 show that the audio data is digitized in the same manner as the video and thus in an embodiment can also be compressed. This compressed audio and video data is then multiplexed as shown in figure 5 with element 43. It is well known to one of ordinary skill in the art that the function of multiplexing generally takes less time than the encoding and compressing step that is being done by Kiriyama. Thus the multiplexing function fits the above

usage of non-time-critical task and the compression/encoding function fits the usage of time-critical task since it is a longer-latency event.

d. Column 1, line 63 – column 2, line 5 show that the invention of Kiriyama achieves excellent efficiency and allows reproduction of synchronous audio and video data so there is no audio-visual gap in this data. This efficiency and audio/video synchronization would have motivated one of ordinary skill in the art to modify the design of Parady to manipulate audio and video data and tasks as taught by Kiriyama.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Parady to manipulate audio and video data and tasks as taught by Kiriyama so that excellent efficiency and synchronous audio and video data may be realized.

34. In regard to claim 24,

a. Parady discloses a processing system as recited in claim 1. Parady has shown in column 2, lines 66-67 that the invention is implemented on an UltraSparc microprocessor. The included Tremblay reference is a document on the structure of the UltraSparc processor and shows in section VII, which starts on page 1661, that there is multimedia support and thus support for video and audio and thus the disclosure of Parady inherently has support for audio and video processing.

b. Parady does not explicitly disclose wherein said processing system is employed in an audio and video encoder/decoder (codec), said audio and video

codec performing compression of audio data and video data to generate a compressed audio stream and a compressed video stream.

c. Kiriyama has disclosed an audio/video multiplexer and demultiplexer system (title) that encodes video and audio (figure 5, elements 37 and 45) as well as decodes video and audio (figure 8, elements 73 and 79). Column 4, lines 37-43 show that the video encoding uses data compression in an embodiment. Column 4, lines 60-64 show that the audio data is digitized in the same manner as the video and thus in an embodiment can also be compressed.

d. Column 1, line 63 – column 2, line 5 show that the invention of Kiriyama achieves excellent efficiency and allows reproduction of synchronous audio and video data so there is no audio-visual gap in this data. This efficiency and audio/video synchronization would have motivated one of ordinary skill in the art to modify the design of Parady to manipulate audio and video data and tasks as taught by Kiriyama.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Parady to manipulate audio and video data and tasks as taught by Kiriyama so that excellent efficiency and synchronous audio and video data may be realized.

35. In regard to claim 25,

a. Parady discloses a processing system as recited in claim 1. Parady has shown in column 2, lines 66-67 that the invention is implemented on an UltraSparc microprocessor. The included Tremblay reference is a document on

the structure of the UltraSparc processor and shows in section VII, which starts on page 1661, that there is multimedia support and thus support for video and audio and thus the disclosure of Parady inherently has support for audio and video processing.

b. Parady does not explicitly disclose wherein said non-time-critical tasks include multiplexing of said compressed audio stream with said compressed video stream, said time-critical tasks including providing video data for compression thereof.

c. Kiriyama has disclosed an audio/video multiplexer and demultiplexer system (title) that encodes video and audio (figure 5, elements 37 and 45) as well as decodes video and audio (figure 8, elements 73 and 79). Column 4, lines 37-43 show that the video encoding uses data compression in an embodiment. Column 4, lines 60-64 show that the audio data is digitized in the same manner as the video and thus in an embodiment can also be compressed. This compressed audio and video data is then multiplexed as shown in figure 5 with element 43. It is well known to one of ordinary skill in the art that the function of multiplexing generally takes less time than the encoding and compressing step that is being done by Kiriyama. Thus the multiplexing function fits the above usage of non-time-critical task and the compression/encoding function fits the usage of time-critical task since it is a longer-latency event.

d. Column 1, line 63 – column 2, line 5 show that the invention of Kiriyama achieves excellent efficiency and allows reproduction of synchronous audio and

video data so there is no audio-visual gap in this data. This efficiency and audio/video synchronization would have motivated one of ordinary skill in the art to modify the design of Parady to manipulate audio and video data and tasks as taught by Kiriyama.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Parady to manipulate audio and video data and tasks as taught by Kiriyama so that excellent efficiency and synchronous audio and video data may be realized.

Conclusion

36. The following is text cited from 37 CFR 1.111(c): In amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or patent owner must clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. The applicant or patent owner must also show how the amendments avoid such references or objections.

37. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. The following references have been disclosed to further show the art with respect to blocked multithreading in general.

US Pat No 6,553,487 to Sukonik uses multiple register files and flows of control (threads) where switching is done between each thread based on a priority of the task context.

US Pat No 6,292,888 to Nemirovsky teaches a processing system with multiple register file groups with different states associated with different streams where one stream facilitates main execution at a time.

US Pat No 5,812,868 to Moyer discloses a method and apparatus for selecting a register file in a processing system depending on the context of the system.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Shane F Gerstl whose telephone number is (703)305-7305. The examiner can normally be reached on M-F 6:45-4:15 (First Friday Off).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703)305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Shane F Gerstl
Examiner
Art Unit 2183

SFG
May 12, 2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100